

```

*****
;
; PROGRAM ID: DOUBLE D BOOTSTRAP (DDBOOT)
;
; *****
;
; VERSION: CP/M 2.2 RELEASE 2
;
; *****
;
; DISTRIBUTOR: JADE COMPUTER PRODUCTS
; 4901 W. ROSECRANS BLVD.
; HAWTHORNE, CALIFORNIA
; 90250, U.S.A.
;
; *****SK*****

```

```

*****
; THE DOUBLE D BOOTSTRAP PROGRAM (DDBOOT) IS USED TO
; INITIATE THE SYSTEM TRACKS LOAD SEQUENCE FROM DRIVE
; A (OR 0) AND TO PROVIDE CONSOLE I/O SUBROUTINES FOR
; THE DISK OPERATING SYSTEM (CP/M). THIS PROM SHOULD
; BE LOCATED AT F000 HEX. THE SOURCE CODE FOR DDBOOT
; CAN BE ASSEMBLED WITH DIGITAL RESEARCH ASSEMBLER
; ASM.COM. MACHINE CODE IS 8080/8085/Z80 COMPATABLE
; *****

```

```

*****
; DDBOOT INJECTION MODULE IS COMMAND COMPATABLE WITH
; THE FOLLOWING WESTERN DIGITAL CONTROLLER CHIPS.
; DOUBLE D USER SWITCH 0 (U0 OR R0) MUST BE SET TO
; INDICATE THE CONTROLLER CHIP DATA BUS POLARITY.
; *****
; CONTROLLER IC USER SW0
; -----
; FD1791-02 (01) CLOSED
; FD1793-02 (01) OPENED
; FD1795-02 CLOSED
; FD1797-02 OPENED
; *****

```

```

; THE FD1795-02 AND FD1797-02 PROVIDE ENHANCED SINGLE
; DENSITY PERFORMANCE IN THAT THESE CHIPS ARE FULLY
; COMPATABLE WITH FD1771-01 3740 FORMATS.
; *****

```

```

*****
; CBIOS SCRATCH ***** SYSTEM MEMORY ALLOCATION
; *****
; ADDRESS NAME FUNCTION
; -----
; 0040-41 D$ADDR ADDRESS POINTER TO DOUBLE D
; 0042 D$MASK STATUS PORT HALT BIT MASK
; 0043 D$TEMP TEMPORARY, INSERT DISK MSG
; *****

```

```

; *****
; MODIFICATION MAY BE NEEDED. THIS PROM PROVIDES THE *
; CONSOLE STATUS, INPUT, AND OUTPUT SUBROUTINES. MANY *
; END USER SYSTEMS REQUIRE UART / USART AND BAUD RATE *
; GENERATOR INITIALIZATION. THESE ROUTINES MAY NEED *
; TO BE PATCHED TO PROVIDE FOR PROPER CONSOLE LINKAGE *
; PATCHING MAY ALSO BE DONE FOR SOME DISK DRIVES. *
; *****
; FROM LOCATIONS THAT MAY NEED PATCHING *
; *****
; SYSTEM INITIALIZATION - UART/USART, BAUD RATE, AND *
; AND POWER ON JUMP CIRCUITRY MAY REQUIRE SOFTWARE *
; INITIALIZATION. A PATCH AREA IS RESERVED AT "INIT" *
; *****
; CONSOLE STATUS CHECK - RETURNS KEYBOARD STATUS TO *
; CP/M OPERATING SYSTEM. THIS SUBROUTINE MUST USE *
; THE CORRECT PORT ADDRESS AND TEST PROPER STATUS BIT.*
; ROUTINE IS LABELED "CNS$CK" AND CONTAINS PATCH AREA.*
; *****
; CONSOLE INPUT - RETURNS KEYBOARD CHARACTER TO CP/M *
; OPERATING SYSTEM. THIS ROUTINE MUST ALSO USE THE *
; CORRECT PORT ADDRESSING. THIS ROUTINE IS LABELED *
; "CNS$IN" AND CONTAINS A PATCH AREA. *
; *****
; CONSOLE OUTPUT - DISPLAYS CHARACTER TO CONSOLE UNIT.*
; THIS ROUTINE MUST USE THE CORRECT PORT ADDRESS FOR *
; BOTH THE OUTPUT STATUS AND OUTPUT DATA PORTS. THIS *
; ROUTINE MUST TEST THE CONSOLE OUTPUT STATUS BIT. *
; THIS ROUTINE IS LABELED "CNS$OT" AND CONTAINS A *
; PATCH AREA. *
; *****
; BOARD REVISION - NOTE SOURCE FILE HAS CONDITIONAL *
; STATEMENTS FOR BOARD REVISION. REVISION B BOARDS *
; MAY BE MODIFIED TO ACT AS REV-C (CALL JADE) OR *
; DDBOOT MAY BE PATCHED. PROM IS FOR REV-C. *
; ----- *
; ADDRESS NAME FUNCTION REV-C REV-B *
; ----- *
; F040 DS$ASW ADDR SW MASK 0E 0C *
; F043 D$BASE ADDR 8K RANGE E0 E4 ** *
; F04B DS$HLT DD HALT BIT 01 02 *
; ----- *
; ** SHOULD BE "E0" IF MA10 JUMPER IS INSTALLED. *
; *****
; DISK DRIVES - DDBOOT IS USING A 10 MILLISECOND STEP *
; WHICH WILL HOME THE R/W HEAD ON MOST DRIVES. IF THE *
; USER DRIVE IS MUCH FASTER (SHUGART SA850 OR SIEMENS *
; FD100-8D) THEN THE STEP RATE CONSTANT MAY BE PATCHED*
; NOTE: A DDBOOT PROM PATCHED FOR FAST DRIVES WILL *
; NOT FUNCTION PROPERLY IF LATER USED WITH SLOWER *
; DRIVES. SLOWER STEPS SHOULD ALWAYS WORK. A DELAY *
; BEFORE READING HAS BEEN PROVIDED FOR HEAD LOAD TIME *
; AND CAN BE USED FOR DRIVE-MOTOR START UP TIME WHEN *
; THE DRIVE MOTORS ARE CONTROLLED BY THE DOUBLE-D. *
; ----- *
; ADDRESS NAME FUNCTION *
; ----- *
; F1CE-F1CF IM$TM$STP STEP TIME *
; F1F7-F1F8 IM$TM$DBR DELAY BEFORE READ *

```

```

; *****
; *****
; DOUBLE D BOOTSTRAP SYSTEM ADDRESS
; *****

F000 = FROM$ADDR EQU 0F000H ;DDBOOT SYSTEM ADDRESS.

; *****
; SET DOUBLE D SYSTEM PORT ADDRESS
; *****

0043 = D$PORT EQU 043H ;DOUBLE D PORT ADDRESS.

; *****
; SET USER DOUBLE D BOARD REVISION
; *****

0001 = TRUE EQU 1 ;SET TRUE TO LOGIC ONE.
0000 = FALSE EQU 0 ;SET FALSE TO LOGIC ZERO
0000 = REV$B EQU FALSE ;SET TRUE FOR REV B BOARDS.
0001 = REV$C EQU TRUE ;SET TRUE FOR REV C BOARDS.
0000 = MA10 EQU FALSE ;TRUE IF MA10 JUMPED (REV-B).

; *****
; DEFINE HALT MASK AND BASE ADDRESS OF DOUBLE D
; *****

IF REV$B
DS$HLT EQU 002H ;STATUS PORT HALT INDICATOR.
DS$ASW EQU 00CH ;STATUS PORT ADDR SW MASK.
D$BASE SET 0E400H ;SYSTEM WINDOW BASE ADDRESS
ENDIF

IF MA10
D$BASE SET 0E000H ;SYSTEM WINDOW BASE ADDRESS
ENDIF

IF REV$C
0001 = DS$HLT EQU 001H ;STATUS PORT HALT INDICATOR.
000E = DS$ASW EQU 00EH ;STATUS PORT ADDR SW MASK.
D$BASE SET 0E000H ;SYSTEM WINDOW BASE ADDRESS
ENDIF

; *****
; BOOTSTRAP LINKAGE ADDRESS.
; *****

0080 = BSTACK EQU 0080H ;BOOTSTRAP TOP OF STACK.
0040 = D$ADDR EQU 0040H ;DOUBLE D ADDRESS POINTER.
0042 = D$MASK EQU 0042H ;DOUBLE D HALT BIT ADDR.
0043 = D$TEMP EQU 0043H ;DDBOOT TEMPORARY LOCATION.
0377 = BL$DCS EQU 0377H ;DCM DISK CONTROLLER STATUS.
0378 = BL$ADR EQU 0378H ;DCM LOAD AND JUMP ADDR PNTR.
037A = BL$BSZ EQU 037AH ;DCM BLOCK LOAD SIZE.
0080 = BL$DNR EQU 0080H ;DRIVE NOT READY BIT.

; *****
; DOUBLE D HARDWARE COMMANDS
; *****

```

```

; *****

0080 =      DC$BGN  EQU      080H      ;RESET Z80A AND EXECUTE.
0001 =      DC$MRQ  EQU      001H      ;REQUEST MEMORY WINDOW.
0000 =      DC$MRT  EQU      000H      ;RELEASE MEMORY WINDOW.
0001 =      DC$MB0  EQU      001H      ;SELECT MEMORY BANK 0.
0003 =      DC$MB1  EQU      003H      ;SELECT MEMORY BANK 1.
0002 =      DC$EXC  EQU      002H      ;ISSUE DOUBLE D INTERRUPT.

; *****
; ASSEMBLER DIRECTIVES
; *****

F000                                ORG      PROM$ADDR      ;MODULE ADDRESS.

; *****
; DDBOOT FUNCTIONS VECTOR TABLE
; *****

F000 C312F0      JMP      INIT      ;INITIALIZE AND BOOT.
F003 C33AF0      JMP      BOOT      ;REBOOT DISK SYSTEM.
F006 C3D7F0      JMP      CNS$CK      ;CONSOLE STATUS.
F009 C3F3F0      JMP      CNS$IN      ;CONSOLE INPUT.
F00C C310F1      JMP      CNS$OT      ;CONSOLE OUTPUT.
F00F C32FF1      JMP      MSG$OT      ;MESSAGE TO CONSOLE.

; *****
; INITIALIZE SYSTEM HARDWARE - USER PATCH AREA
; *****

F012 00000000    INIT:      NOP!NOP!NOP!NOP      ;PATCH AREA.
F016 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F01A 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F01E 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F022 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F026 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F02A 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F02E 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F032 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.
F036 00000000      NOP!NOP!NOP!NOP      ;PATCH AREA.

; *****
; SET STACK AND DETERMINE CONTROLLER ADDRESS
; *****

F03A 318000      BOOT:      LXI      SP,BSTACK      ;SET STACK POINTER.
F03D DB43          IN      D$PORT      ;INPUT STATUS PORT.
F03F E60E          ANI      DS$ASW      ;MASK FOR ADDR SWS.
F041 07           RLC          ;POSITION BITS.
F042 F6E0          ORI      D$BASE SHR 8      ;OR IN BASE ADDR.
F044 67           MOV      H,A          ;HIGH BYTE VALUE.
F045 2E00          MVI      L,0          ;LOW BYTE VALUE.
F047 224000        SHLD     D$ADDR      ;STORE THE ADDRESS
F04A 3E01          MVI      A,DS$HLT      ;LOAD HALT BIT MASK.
F04C 324200        STA      D$MASK      ;STORE FOR BIOS USE.
F04F 324300        STA      D$TEMP      ;SET REPEAT FLAG NZ.

; *****
; INJECT BOOT MODULE INTO CONTROLLER
; *****

```



```

F052 3E01      INJECT: MVI      A,DC$MBO      ;REQUEST DD MEM BANK 0.
F054 D343      OUT        D$PORT      ;ISSUE COMMAND.
F056 01C800     LXI        B,IM$END-IM$BGN ;INJECTION MODULE SIZE.
F059 EB        XCHG         ;D$ADDR HL TO DE.
F05A 2186F1     LXI        H,IM$BGN      ;INJECTION MODULE ADDR.
F05D CDA4F0     CALL       BLOCK        ;BLOCK MOVE.

```

```

; *****
; RESET AND START THE DISK PROCESSOR
; *****

```

```

F060 3E80      MVI        A,DC$BGN      ;BEGIN DD PROCESSOR.
F062 D343      OUT        D$PORT      ;ISSUE COMMAND.
F064 E3        XTHL         ;ALLOW DOUBLE D TIME
F065 E3        XTHL         ;TO START UP.

```

```

; *****
; WAIT FOR TASK COMPLETION
; *****

```

```

F066 3A4200     LDA        D$MASK      ;HALT BIT MASK.
F069 47        MOV        B,A          ;MOVE INTO B REG.
F06A DB43      WAIT: IN      D$PORT      ;INPUT DD STATUS.
F06C A0        ANA        B            ;TEST HALT* STATUS.
F06D C26AF0     JNZ        WAIT        ;WAIT TILL HALTED.

```

```

; *****
; SWITCH CONTROLLER MEMORY INTO SYSTEM BUS
; *****

```

```

F070 3E01      MVI        A,DC$MRQ      ;REQUEST MEM (BANK 0).
F072 D343      OUT        D$PORT      ;ISSUE COMMAND.

```

```

; *****
; CHECK FOR BOOTSTRAP MALFUNCTION
; *****

```

```

F074 2A4000     LHLD       D$ADDR      ;CONTROLLER ADDRESS.
F077 117703     LXI        D,BL$DCS    ;ERROR CODE IM$BGN.
F07A 19        DAD        D            ;SET HL POINTER.
F07B 7E        MOV        A,M          ;GET ERROR CODE.
F07C E680      ANI        BL$DNR      ;TEST DRIVE NOT READY.
F07E C2B1F0     JNZ        INSERT      ;IF DRIVE NOT READY.
F081 7E        MOV        A,M          ;GET ERROR CODE.
F082 A7        ANA        A            ;TEST REGISTER.
F083 C2C5F0     JNZ        BAD$LD      ;BAD LOAD.

```

```

; *****
; PERFORM BLOCK TRANSFER FROM DISK MEMORY
; *****

```

```

F086 2A4000     LHLD       D$ADDR      ;CONTROLLER ADDRESS.
F089 117803     LXI        D,BL$ADR    ;LOAD ADDRESS PNTR.
F08C 19        DAD        D            ;SET HL POINTER.
F08D 5E        MOV        E,M          ;LOW ORDER ADDR.
F08E 23        INX        H            ;INCREMENT HL.
F08F 56        MOV        D,M          ;HIGH ORDER ADDR.
F090 23        INX        H            ;REQUIRES BL.BSZ NEXT.
F091 4E        MOV        C,M          ;LOW ORDER LENGTH.

```

```

F092 23          INX      H          ;INCREMENT HL.
F093 46          MOV      B,M        ;HIGH ORDER LENGTH.
F094 D5          PUSH     D          ;USE AS JUMP ADDR.
F095 3E03        MVI      A,DC$MB1   ;SWITCH TO MEM BANK 1.
F097 D343        OUT      D$PORT     ;ISSUE COMMAND.
F099 2A4000      LHLD     D$ADDR     ;DOUBLE D MEM ADDRESS.
F09C CDA4F0      CALL     BLOCK      ;MOVE BIOS MODULE.

```

```

;*****
; TRANSFER CONTROL TO OPERATING SYSTEM *
;*****

```

```

F09F 3E01        MVI      A,DC$MBO   ;SWITCH TO BANK.0
FOA1 D343        OUT      D$PORT     ;ISSUE COMMAND.
FOA3 C9          RET                ;GOTO BIOS COLD ENTRY.

```

```

;*****
; BLOCK MOVE SUBROUTINE (Z80 BLOCK MOVE REGISTERS) *
;*****

```

```

FOA4 7E          BLOCK:  MOV      A,M          ;GET BYTE.
FOA5 23          INX      H          ;INC POINTER
FOA6 EB          XCHG                     ;GET DESTINATION.
FOA7 77          MOV      M,A          ;PUT BYTE.
FOA8 23          INX      H          ;INC POINTER
FOA9 EB          XCHG                     ;GET SOURCE.
FOAA 0B          DCX      B          ;ONE LESS TO DO.
FOAB 78          MOV      A,B          ;GET HI COUNT.
FOAC B1          ORA      C          ;GET LO COUNT.
FOAD C2A4F0      JNZ      BLOCK        ;FINISH LOADING.
FOB0 C9          RET

```

```

;*****
; DISK DRIVE IS NOT READY *
;*****

```

```

FOB1 3A4300      INSERT: LDA      D$TEMP     ;LOAD INIT FLAG.
FOB4 A7          ANA      A          ;TEST FOR INITIAL.
FOB5 CA52F0      JZ       INJECT          ;TRY BOOTING AGAIN.
FOB8 AF          XRA      A          ;ZERO A REGISTER.
FOB9 324300      STA      D$TEMP     ;CLEAR INITIAL FLAG.
FOBC 2153F1      LXI      H,MSG$IN    ;INSERT MESSAGE ADDR.
FOBF CD2FF1      CALL     MSG$OT      ;OUTPUT MESSAGE.
FOC2 C352F0      JMP      INJECT      ;TRY BOOTING AGAIN.

```

```

;*****
; DOUBLE D BOOTSTRAP MALFUNCTION *
;*****

```

```

FOC5 324300      BAD$LD: STA      D$TEMP     ;STORE ERROR CODE.
FOC8 216EF1      LXI      H,MSG$ER     ;ERROR MESSAGE ADDEESS.
FOCB CD2FF1      CALL     MSG$OT      ;DISPLAY MESSAGE.
FOCE 3A4300      LDA      D$TEMP     ;LOAD ERROR CODE.
FOD1 CD3BF1      CALL     HXB$OT      ;DISPLAY HEX BYTE.
FOD4 760000      HLT!NOP!NOP          ;HALT OR JUMP MONITOR.

```

```

;*****
; CONSOLE INPUT AND OUTPUT *
;*****
;
;

```

```

; XXX$SP: STATUS PORT ADDRESS *
; XXX$SB: STATUS READY BIT *
; XXX$SI: IF READY TRUE IS "1" USE "00" ELSE "FF" *
; XXX$DP: DATA PORT ADDRESS *
; *
; *****

0000 = CNI$SP EQU 000H ;INPUT STATUS PORT.
0002 = CNI$SB EQU 002H ;INPUT STATUS BIT.
0000 = CNI$SI EQU 000H ;INPUT STATUS INVERT.
0001 = CNI$DP EQU 001H ;INPUT DATA PORT.

0000 = CNO$SP EQU 000H ;OUTPUT STATUS PORT.
0004 = CNO$SB EQU 004H ;OUTPUT STATUS BIT.
0000 = CNO$SI EQU 000H ;OUTPUT STATUS INVERT.
0001 = CNO$DP EQU 001H ;OUTPUT DATA PORT.

; *****
; CONSOLE INPUT STATUS CHECK *
; *****

F0D7 DB00 CNS$CK: IN CNI$SP ;INPUT STATUS PORT.
F0D9 EE00 XRI CNI$SI ;ADJUST POLARITY.
F0DB E602 ANI CNI$SB ;TEST READY BIT.
F0DD C8 RZ ;ZERO IS NOT READY.
F0DE 3EFF MVI A,0FFH ;SET CONSOLE READY.
F0E0 C9 RET ;ONES INDICATE READY.

F0E1 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.
F0E7 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.
F0ED 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.

; *****
; CONSOLE DATA INPUT *
; *****

F0F3 CDD7F0 CNS$IN: CALL CNS$CK ;TEST INPUT READY.
F0F6 CAF3F0 JZ CNS$IN ;REPEAT TEST FOR RDY.
F0F9 DB01 IN CNI$DP ;INPUT CONSOLE DATA.
F0FB E67F ANI 07FH ;SEVEN BITS OF ASCII.
F0FD C9 RET ;RETURN WITH DATA.

F0FE 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.
F104 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.
F10A 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.

; *****
; CONSOLE DATA OUTPUT *
; *****

F110 DB00 CNS$OT: IN CNO$SP ;OUTPUT STATUS PORT.
F112 EE00 XRI CNO$SI ;ADJUST POLARITY.
F114 E604 ANI CNO$SB ;TEST READY BIT.
F116 CA10F1 JZ CNS$OT ;TEST AGAIN FOR RDY.
F119 79 MOV A,C ;OUTPUT SETUP.
F11A D301 OUT CNO$DP ;OUTPUT CONSOLE DATA.
F11C C9 RET ;RETURN COMPLETE.

F11D 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.
F123 0000000000 DB 0,0,0,0,0,0 ;PATCHING AREA.

```

```
F129 0000000000      DB      0,0,0,0,0,0      ;PATCHING AREA.
```

```
*****
; MESSAGE DISPLAY ROUTINE - HL REG POINTS TO STRING *
*****
```

```
F12F 7E      MSG$OT: MOV      A,M      ;LOAD CHARACTER/BYTE.
F130 FE24      CPI      '$'      ;CHECK FOR TERMINATOR.
F132 C8      RZ      ;EXIT IF TERMINATOR.
F133 4F      MOV      C,A      ;PASS BYTE IN C REG.
F134 CD10F1     CALL     CNS$OT      ;DISPLAY CHARACTER.
F137 23      INX      H      ;POINT TO NEXT BYTE.
F138 C32FF1     JMP      MSG$OT      ;REPEAT SEQUENCE.
```

```
*****
; DISPLAY A REGISTER IS HEXIDECIMAL *
*****
```

```
F13B F5      HXB$OT: PUSH     PSW      ;SAVE A REGISTER.
F13C 0F0F0F0F RRC!RRC!RRC!RRC      ;SHIFT 4 PLACES.
F140 CD44F1     CALL     HXN$OT      ;DISPLAY HEX NIBBLE.
F143 F1      POP      PSW      ;RESTORE A REGISTER.
F144 E60F      HXN$OT: ANI      00FH      ;MASK LOWER NIBBLE.
F146 FE0A      CPI      00AH      ;TEST IF LETTER HEX.
F148 DA4DF1     JC      HXN$NM      ;DISPLAY NUMBER.
F14B C607      ADI      'A'-'9'-1      ;ADD LETTER OFFSET.
F14D C630      HXN$NM: ADI      '0'      ;START WITH ASCII 0.
F14F 4F      MOV      C,A      ;OUTPUT BYTE TO C REG.
F150 C310F1     JMP      CNS$OT      ;CONSOLE OUTPUT.
```

```
*****
; SYSTEM BOOTSTRAP MESSAGE AREA *
*****
```

```
F153 0D0A0A494EMSG$IN: DB      CR,LF,LF,'INSERT SYSTEM DISKETTE ','$'
F16E 0D0A0A4444MSG$ER: DB      CR,LF,LF,'DDBOOT LOAD ERROR - ','$'
```

```
000D =      CR      EQU      00DH      ;CARRAIGE RETURN.
000A =      LF      EQU      00AH      ;LINE FEED COMMAND.
```

```
*****
*****
; INJECTION MODULE *** THE FOLLOWING EXECUCUTES IN DD *
*****
; THIS SECTION OF CODE IS HAS BEEN WRITTEN WITH AN *
; ADDRESS OFFSET SO AS TO ASSEMBLE WITH IM$BGN AT *
; LOCATION ZERO. *
*****
*****
```

```
*****
; DOUBLE D INTERNAL PORT ASSIGNMENTS *
*****
```

```
0000 =      IM$BL$STS      EQU      000H      ;BOARD STATUS
0000 =      IM$BL$CTL      EQU      000H      ;BOARD CONTROLS
0004 =      IM$WD$CMD      EQU      004H      ;179X COMMAND REGISTER
0004 =      IM$WD$STS      EQU      004H      ;179X STATUS REGISTER
```

```

0005 = IM$WD$TRK EQU 005H ;179X TRACK REGISTER
0006 = IM$WD$SEC EQU 006H ;179X SECTOR REGISTER
0007 = IM$WD$DTA EQU 007H ;179X DATA REGISTER
0008 = IM$XP$STP EQU 008H ;STEPPER PULSE
0010 = IM$XP$MTO EQU 010H ;MOTOR TIME OUT
0040 = IM$XP$MTX EQU 040H ;MOTOR TIME EXTEND
0080 = IM$XP$DSH EQU 080H ;DATA SYNC HOLD

; *****
; 179X-02 COMMAND CODES *
; *****

0018 = IM$DC$HDL EQU 00011000B ;LOAD R/W HEAD.
0098 = IM$DC$RMS EQU 10011000B ;READ MULTI-SECTOR.
00D0 = IM$DC$STS EQU 11010000B ;SET TYPE 1 STATUS.

; *****
; Z80 INSTRUCTION HEX CODES - NOTE HI/LOW ORDER SWAP *
; *****

21FD = IM$LXIY EQU 021FDH ;LOAD Y REG IMED.
45ED = IM$RETN EQU 045EDH ;RETN (NMI RETURN).
E3FD = IM$XTIY EQU 0E3FDH ;EXCHANGE (SP) <> IY.

; *****
; BOARD STATUS AND CONTROL PORTS *
; *****

0001 = IM$BS$USO EQU 001H ;179X-02 POLARITY TEST.
0004 = IM$BC$DRO EQU 004H ;DRIVE 0 SELECT/ENABLE.

; *****
; DISK STATUS MASKS *
; *****

009C = IM$DM$RER EQU 10011100B ;READ ERROR TEST MASK.
0004 = IM$DM$TKO EQU 00000100B ;TRACK 0 TEST.
0080 = IM$DM$DNR EQU 10000000B ;DRIVE NOT READY.

; *****
; DISK DRIVE PARAMETERS *
; *****

000A = IM$TM$STP EQU 10 ;STEPPER INTERVAL - MS.
0028 = IM$TM$DBR EQU 40 ;DELAY BEFORE READ- MS.
004C = IM$NB$TRK EQU 76 ;MAXIMUM NMBR OF STEPS.

; *****
; INTERNAL MEMORY ASSIGNMENTS *
; *****

0000 = IM$BKO EQU 0000H ;LOWER BANK ADDRESS.
0400 = IM$BKL EQU 0400H ;1K BANK LENGTH.
0400 = IM$BK1 EQU IM$BKO+IM$BKL ;UPPER BANK ADDRESS.
0066 = IM$NMI EQU IM$BKO+0066H ;NON-MASKABLE INT ADDR.
0376 = IM$BL$ERC EQU IM$BKO+0376H ;ERROR CODE LOCATION.
0377 = IM$BL$DCS EQU IM$BKO+BL$DCS ;DISK CONTROLLER STAT.

; *****
; BOOTSTRAP COMMUNICATION *

```



```

; *****
0001 = BE$HOM EQU 001H ;HOME ERROR.
0002 = BE$RDA EQU 002H ;READ ERROR A.
0004 = BE$RDB EQU 004H ;READ ERROR B.

; *****
; DISK CONTROLLER MODULE (DCM) LINKAGE *
; *****

000D = DCM$SS EQU 13 ;FIRST DCM SECTOR = 13.
0403 = DCM$BG EQU IM$BK1+3 ;DCM COLD START ENTRY.
0400 = DCM$LN EQU 0400H ;DCM LENGTH

; *****
; SET STACK, START DRIVE MOTOR, AND SET INVERT SW (C) *
; *****

F186 310004 IM$BGN: LXI SP,IM$BK1 ;SET UP STACK.
F189 DB40 IN IM$XP$MTX ;TURN ON MOTOR.
F18B 0E00 MVI C,0 ;ASSUME 1793.
F18D DB00 IN IM$BL$STS ;INPUT STATUS.
F18F E601 ANI IM$BS$USO ;TEST USER SW 0.
F191 C21000 JNZ IM$DRV-IM$BGN ;GOTO SELECT DRV.
F194 0EFF MVI C,OFFH ;1791-01 INVERTS.

; *****
; CLEAR 179X-01 INTERRUPT AND SELECT DRIVE 0 *
; *****

F196 CD5000 IM$DRV: CALL IM$STS-IM$BGN ;179X-01 FORCED CLEAR.
F199 3E04 MVI A,IM$BC$DRO ;DRIVE 0, ENABLED.
F19B D300 OUT IM$BL$CTL ;OUTPUT CONTROLS.

; *****
; CHECK FOR DRIVE READY SIGNAL *
; *****

F19D CD5000 CALL IM$STS-IM$BGN ;GET DRIVE STATUS.
F1A0 327703 STA IM$BL$DCS ;STORE DRIVE STATUS.
F1A3 E680 ANI IM$DM$DNR ;CHECK DRIVE NOT RDY.
F1A5 CA2600 JZ IM$HDL-IM$BGN ;IF READY, BOOT SYSTEM.
F1A8 AF XRA A ;ZERO A REGISTER.
F1A9 C3B100 JMP IM$HLT-IM$BGN ;DOUBLE D SHUTDOWN.

; *****
; LOAD R/W HEAD ON SELECTED DRIVE *
; *****

F1AC 79 IM$HDL: MOV A,C ;GET TRACK 0 VALUE.
F1AD D305 OUT IM$WD$TRK ;SET TRACK REGISTER.
F1AF D307 OUT IM$WD$DTA ;SEEK SAME TRACK.
F1B1 FD21 DW IM$LXIY ;Z80 LXIY HEX CODE.
F1B3 3700 DW IM$HME-IM$BGN ;SET NMI RETURN ADDR.
F1B5 3E18 MVI A,IM$DC$HDL ;HEAD LOAD COMMAND.
F1B7 A9 XRA C ;INVERT (1791-01).
F1B8 D304 OUT IM$WD$CMD ;ISSUE COMMAND.
F1BA C33400 IM$WFI: JMP IM$WFI-IM$BGN ;WAIT FOR INTERRUPT.

; *****

```

```

; POSITION R/W HEAD AT TRACK ZERO
; *****

```

```

F1BD 2E4C      IM$HME: MVI      L,IM$NB$TRK      ;SET MAX TRACKS.
F1BF CD5000    IM$STP: CALL     IM$STS-IM$BGN     ;GET 179X STATUS.
F1C2 E604      ANI          IM$DM$TKO          ;TEST TRACK 0 BIT.
F1C4 C27000    JNZ          IM$RSU-IM$BGN     ;TRACK 0 EXIT.
F1C7 2D        DCR          L                  ;DEC ATTEMPTS.
F1C8 CAAF00    JZ           IM$EHM-IM$BGN     ;CANT FIND TRK 0?
F1CB DB08      IN           IM$XP$STP          ;ISSUE STEP PULSE.
F1CD 110A00    LXI          D,IM$TM$STP        ;STEP INTERVAL TIME.
F1D0 CDBA00    CALL         IM$TMR-IM$BGN     ;PAUSE FOR PERIOD.
F1D3 C33900    JMP          IM$STP-IM$BGN     ;TRY ANOTHER TIME.

```

```

; *****
; GET UPDATED 179X-01 STATUS
; *****

```

```

F1D6 3ED0      IM$STS: MVI      A,IM$DC$STS      ;TYPE 4 - STATUS.
F1D8 A9        XRA          C                  ;INVERT (1791-01).
F1D9 D304      OUT          IM$WD$CMD          ;ISSUE COMMAND.
F1DB E3        XTHL         ;DELAY
F1DC E3        XTHL         ;DELAY
F1DD E3        XTHL         ;DELAY
F1DE E3        XTHL         ;DELAY
F1DF DB04      IN           IM$WD$STS          ;GET STATUS
F1E1 A9        XRA          C                  ;INVERT (1791-01).
F1E2 C9        RET          ;RETURN TO CALLER.

```

```

; *****
; DISK INTERRUPT "NMI" ROUTINE
; *****

```

```

F1EC                      ORG      IM$BGN+IM$NMI

F1EC DB04      IN           IM$WD$STS          ;GET 179X STATUS.
F1EE A9        XRA          C                  ;INVERT (1791-01).
F1EF 327703    STA          IM$BL$DCS          ;MAKE STATUS VISIBLE.
F1F2 FDE3      DW          IM$XTIY            ;EXCHANGE (SP) <> IY.
F1F4 ED45      DW          IM$RETN            ;NMI RETURN (RETN).

```

```

; *****
; SET-UP FOR DCM READ OPERATION
; *****

```

```

F1F6 112800    IM$RSU: LXI      D,IM$TM$DBR      ;DELAY BEFORE READ.
F1F9 CDBA00    CALL         IM$TMR-IM$BGN     ;CALL MS. TIMER.
F1FC 110004    LXI          D,IM$BKL          ;SET BANK LENGTH
F1FF 210004    LXI          H,IM$BK1          ;DCM LOAD ADDRESS
F202 FD21      DW          IM$LXIY            ;Z80 LXI Y HEX CODE.
F204 A500      DW          IM$REA-IM$BGN     ;READ ERROR TRAP.
F206 3E0D      MVI          A,DCM$SS          ;FIRST SEC OF DCM.
F208 A9        XRA          C                  ;INVERT (1791-01)
F209 D306      OUT          IM$WD$SEC          ;SET 179X SEC REG.
F20B 3E98      MVI          A,IM$DC$RMS       ;READ MULTI-SECTOR.
F20D A9        XRA          C                  ;INVERT (1791-01).
F20E D304      OUT          IM$WD$CMD          ;ISSUE 179X COMMAND.

```

```

; *****
; ACCEPT EACH BYTE AND STORE IN MEMORY
; *****

```

```

; *****
F210 DB80      IM$RBT: IN      IM$XP$DSH      ;WAIT FOR DATA.
F212 DB07      IN      IM$WD$DTA      ;INPUT INV DATA.
F214 A9        XRA      C              ;INVERT (1791-01).
F215 77        MOV      M,A          ;STORE DCM BYTE.
F216 23        INX      H              ;INCREMENT POINTER.
F217 1B        DCX      D              ;DECREMENT LENGTH.
F218 7A        MOV      A,D          ;GET HIGH REG.
F219 B3        ORA      E              ;THEN OR-IN LOW REG.
F21A C28A00     JNZ      IM$RBT-IM$BGN ;READ ANOTHER BYTE.

```

```

; *****
; TEST READ STATUS, TERMINATE OPERATION, GO DCM      *
; *****

```

```

F21D DB04      IM$TRS: IN      IM$WD$STS      ;INPUT READ STATUS.
F21F A9        XRA      C              ;INVERT (1791-01).
F220 E69C      ANI      IM$DM$RER      ;TEST FOR ERRORS.
F222 C2AA00     JNZ      IM$REB-IM$BGN      ;READ ERROR TRAP.
F225 CD5000     CALL     IM$STS-IM$BGN      ;TERMINATE READ.
F228 C30304     JMP      DCM$BG          ;TRANSFER TO DCM.

```

```

; *****
; READ ERROR HAS BEEN DETECTED      *
; *****

```

```

F22B 3E02      IM$REA: MVI      A,BE$RDA      ;LOAD READ ERROR CODE.
F22D C3B100     JMP      IM$HLT-IM$BGN      ;GO TO ERROR HALT.
F230 3E04      IM$REB: MVI      A,BE$RDB      ;LOAD READ ERROR CODE.
F232 C3B100     JMP      IM$HLT-IM$BGN      ;GO TO ERROR HALT.
F235 3E01      IM$EHM: MVI      A,BE$HOM      ;HOME ERROR CODE.
F237 327603     IM$HLT: STA      IM$BL$ERC      ;DISPLAY ERROR CODE.
F23A AF        XRA      A              ;ZERO A REG.
F23B D300      OUT      IM$BL$STS      ;DESELECT DRIVE.
F23D DB10      IN      IM$XP$MTO      ;MOTOR OFF!
F23F 76        HLT                      ;TERMINATE.

```

```

; *****
; TIMER - WAIT FOR (BC * 1.0) MILLISECONDS      *
; *****

```

```

F240 3EDC      IM$TMR: MVI      A,220        ;LOAD INT MS VALUE.
F242 3D        IM$TMX: DCR      A              ;DEC FOR 1 MS.
F243 00        NOP                      ;EXTRA TIMING DELAY.
F244 C2BC00     JNZ      IM$TMX-IM$BGN      ;REPEAT FOR 1 MS.
F247 1B        DCX      D              ;TEST FOR ANOTHER MS.
F248 7A        MOV      A,D              ;CHECK REG D.
F249 B3        ORA      E              ;AND REGISTER E.
F24A C2BA00     JNZ      IM$TMR-IM$BGN      ;DO ANOTHER 1 MS.
F24D C9        RET                      ;TIME PERIOD EXPIRED!

```

```

; *****

```

```

F24E          IM$END: END      BEGIN      ;END OF ASSEMBLY.

```

*

DF000,F2FF

```
F000 C3 12 F0 C3 3A F0 C3 D7 F0 C3 F3 F0 C3 10 F1 C3 .....:.....
F010 2F F1 DB 7F 00 00 00 00 00 00 00 00 00 00 00 00 /.....
F020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
F030 00 00 00 00 00 00 00 00 00 00 00 31 80 00 DB 43 E6 .....1...C.
F040 0E 07 F6 E0 67 2E 00 22 40 00 3E 01 32 42 00 32 ....G.."@.>.2B.2
F050 43 00 3E 01 D3 43 01 C8 00 EB 21 86 F1 CD A4 F0 C.>..C....!.....
F060 3E 80 D3 43 E3 E3 3A 42 00 47 DB 43 A0 C2 6A F0 >..C...:B.G.C..J.
F070 3E 01 D3 43 2A 40 00 11 77 03 19 7E E6 80 C2 B1 >..C*@..W..^....
F080 F0 7E A7 C2 C5 F0 2A 40 00 11 78 03 19 5E 23 56 .^.....*@..X..^#V
F090 23 4E 23 46 D5 3E 03 D3 43 2A 40 00 CD A4 F0 3E #N#F.>..C*@.....>
FOA0 01 D3 43 C9 7E 23 EB 77 23 EB 0B 78 B1 C2 A4 F0 ..C.^#.W#.X....
F0B0 C9 3A 43 00 A7 CA 52 F0 AF 32 43 00 21 53 F1 CD .:C...R..2C.!S..
F0C0 2F F1 C3 52 F0 32 43 00 21 6E F1 CD 2F F1 3A 43 /..R.2C.!N../.:C
F0D0 00 CD 3B F1 76 00 00 DB 00 EE 00 E6 02 C8 3E FF ..;.V.....>.
F0E0 C9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
F0F0 00 00 00 CD D7 F0 CA F3 F0 DB 01 E6 7F C9 00 00 .....
F100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
F110 DB 00 EE 00 E6 04 CA 10 F1 79 D3 01 C9 00 00 00 .....Y.....
F120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 7E .....^
F130 FE 24 C8 4F CD 10 F1 23 C3 2F F1 F5 0F 0F 0F 0F .$.0...#./.....
F140 CD 44 F1 F1 E6 0F FE 0A DA 4D F1 C6 07 C6 30 4F .D.....M....00
F150 C3 10 F1 0D 0A 0A 49 4E 53 45 52 54 20 53 59 53 .....INSERT SYS
F160 54 45 4D 20 44 49 53 4B 45 54 54 45 20 24 0D 0A TEM DISKETTE $.
F170 0A 44 44 42 4F 4F 54 20 4C 4F 41 44 20 45 52 52 .DDBOOT LOAD ERR
F180 4F 52 20 2D 20 24 31 00 04 DB 40 0E 00 DB 00 E6 OR - $1...@.....
F190 01 C2 10 00 0E FF CD 50 00 3E 04 D3 00 CD 50 00 .....P.>....P.
F1A0 32 77 03 E6 80 CA 26 00 AF C3 B1 00 79 D3 05 D3 2W....&.....Y...
F1B0 07 FD 21 37 00 3E 18 A9 D3 04 C3 34 00 2E 4C CD ..!7.>.....4..L.
F1C0 50 00 E6 04 C2 70 00 2D CA AF 00 DB 08 11 0A 00 P....P.-.....
F1D0 CD BA 00 C3 39 00 3E D0 A9 D3 04 E3 E3 E3 E3 DB ....9.>.....
F1E0 04 A9 C9 00 00 00 00 00 00 00 00 00 00 DB 04 A9 32 .....2
F1F0 77 03 FD E3 ED 45 11 28 00 CD BA 00 11 00 04 21 W....E.(.....!
F200 00 04 FD 21 A5 00 3E 0D A9 D3 06 3E 98 A9 D3 04 ...!...>....>....
F210 DB 80 DB 07 A9 77 23 1B 7A B3 C2 8A 00 DB 04 A9 .....W#.Z.....
F220 E6 9C C2 AA 00 CD 50 00 C3 03 04 3E 02 C3 B1 00 .....P....>....
F230 3E 04 C3 B1 00 3E 01 32 76 03 AF D3 00 DB 10 76 >....>.2V.....V
F240 3E DC 3D 00 C2 BC 00 1B 7A B3 C2 BA 00 C9 FF FF >.=.....Z.....
F250 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F260 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F270 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F280 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F290 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
F2F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
```

-

